Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/GB05/001009

International filing date: 17 March 2005 (17.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: GB
Number: 0407385.4
Filing date: 31 March 2004 (31.03.2004)

Date of receipt at the International Bureau: 09 May 2005 (09.05.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.
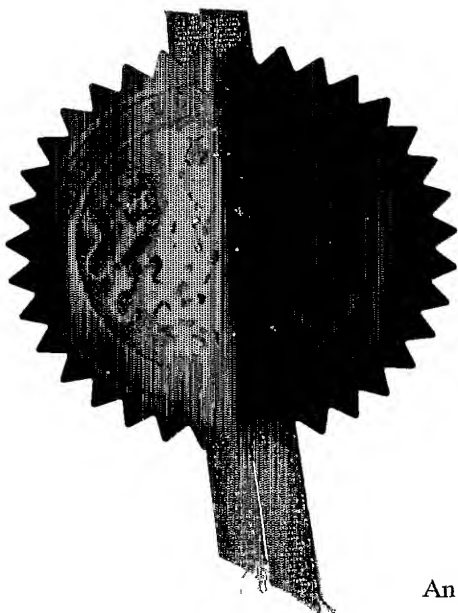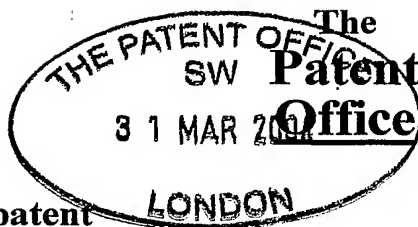
Signed *William Morell*

Dated 12 April 2005

THE PATENT OFFICE SW **The Patent Office**
31 MAR 2004
LONDON

1/77

01APR04 E885611-1 003052
P01/7700 0.00-0407385.4 ACCOUNT CHA

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

1.	Your reference	**A30468**

2.	Patent application number *(The Patent Office will fill in this part)*	**0407385 4**	**3 1 MAR 2004**

3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	**BRITISH TELECOMMUNICATIONS public limited company** **81 NEWGATE STREET** **LONDON, EC1A 7AJ, England** **Registered in England: 1800000**
	Patents ADP number *(if you know it)*	~~4867002~~ *6300038800 1*
	If the applicant is a corporate body, give the country/state of its incorporation	**UNITED KINGDOM**
4.	Title of the invention	**PATHFINDING SYSTEM**

5.	Name of your agent *(if you have one)*	
	"Address for Service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*	**BT GROUP LEGAL** **INTELLECTUAL PROPERTY DEPARTMENT** **PPC5A, BT CENTRE** **81 NEWGATE STREET** **LONDON, EC1A 7AJ, ENGLAND**
	Patents ADP number *(if you know it)*	~~1867004~~ *8772378001*

		Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number			

		Number of earlier application	Date of filing *(day/month/year)*
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application		

8.	Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:* a) *any applicant named in part 3 is not an inventor, or* b) *there is an inventor who is not named as an applicant, or* c) *any named applicant is a corporate body.* *(See note (d))*	**YES**

Patents Form 1/77

9. Enter the number of sheets for any of the
   following items you are filing with this form.
   Do not count copies of the same document

   Continuation sheets of this form   -

   Description   - **19**

   Claim(s)   - **04**

   Abstract   - **01**

   Drawing(s)   - **13** + 13 ✓

10. If you are also filing any of the following,
    state how may against each item

    Priority Documents

    Translations of priority documents

    Statement of inventorship and right
    to grant of a patent *(Patents Form 7/77)*

    Request for preliminary examination   YES (1)
    and search *(Patents Form 9/77)*

    Request for substantive examination
    (Patents Form 10/77)

    Any other documents
    (please specify)

11. e

    I/We request the grant of a patent on the basis of this application.
    Signature(s)   Date:

    Sim Will   **31 March 2004**

    WILLIAMSON, Simeon Paul, Authorised Signatory

12. Name and daytime telephone number of
    person to contact in the United Kingdom   **Mark WATSON**   **020 7356 6163**

Warning
After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes
a)   *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b)   *Write your answers in capital letters using black ink or you may type them.*

c)   *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d)   *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e)   *Once you have filled in the form you must remember to sign and date it.*

f)   *For details of the fee and ways to pay please contact the Patent Office.*

PATHFINDING SYSTEM

TECHNICAL FIELD

5

The invention relates to a method and system for determining a path along a plurality of points in a representation of a virtual world.

INTRODUCTION

10

Systems in which 3-dimensional virtual reality environments (virtual worlds) are defined and created are becoming increasingly prevalent in the current technological age. Such computer systems are applicable in numerous commercial applications, such as for online tourist sites, fantasy worlds, gaming, architectural walk-throughs, estate agency (virtual tours of houses

15   for sale), and many others. Typically, a virtual world might be defined by multiple x,y,z, co-ordinate sets which together map out an environment in three dimensions (x, y, and z axes). Objects and walls (or other items which exist within the virtual world), which a user navigating through the virtual world is not allowed to walk through, can be implemented by what is termed a collisionable mesh. The collisionable mesh may therefore comprise a further set of

20   three dimensional co-ordinates defining shapes within the virtual world which the user must navigate around.

Navigation is required within virtual worlds for any entity passing through that world, such as a computer generated person, vehicle or gaming character. This is typically under the control

25   of an external human user. Traditionally, many commercial implementations of virtual environments have been built on the basis of the self-navigation techniques inherited from games engines, where exploration of all areas of an unknown virtual world is expected, and also a high degree of user competency is assumed. However, as the application of virtual worlds spreads into other commercial areas, it is important that the functionality of navigation

30   evolves to suit the competency and requirements of a whole new wider group of users. Such users may not be as proficient in the navigation around virtual worlds, and furthermore may be disinclined to spend large amounts of time familiarising themselves with the environment. They are likely to become disorientated or bored much more quickly, or even miss important content entirely, unless the user experience is significantly improved.

35

In our day-to-day lives, humans typically employ various techniques to help us with our navigation when moving from place to place. We retain an idea of our location and orientation using fixed points of reference (landmarks), measures of distance travelled and changes in direction so as to prevent ourselves becoming disorientated. Such techniques

5   may not be available to the novel user in a virtual world which they are unfamiliar, particularly if teleporting (e.g. instantaneously transferring between two different points in the world) is applied, as is the case in many gaming environments. Consequently, it is advantageous to provide an automated pathfinding system for the inexperienced user to help them navigate them through the virtual world, allowing them to overcome the problems of lack of familiarity

10   and any frustrations with the traditional interface of self-navigation. Consequently, many systems have been developed which automatically calculate the best path through a virtual world, for example to find the shortest path to view a particular object or to pass through a room.

15   **PRIOR ART**

Article entitled *"Smart Moves: Intelligent Pathfinding"* by Bryan Stout, published in Game Developer, October 1996, and also available online at http://www.gamasutra.com/features/19970801/pathfinding.htm provides an broad
20   introduction to various pathfinding techniques which can be used in virtual worlds. In particular, it discusses and compares breadth-first pathfinding (including Dijkstra's Algorithm), depth-first and best-first pathfinding techniques.

All pathfinding algorithms work by expanding possible paths, one node at a time until a path
25   from the start node to the destination node is found. The order in which the nodes are attempted is critical to the speed and quality (e.g. length) of the final path that is found. In a breadth-first techniques, the algorithm searches firstly through all a nodes' immediate neighbours, and then moving onto the neighbours of those neighbouring nodes. In contrast, in depth-first techniques, the algorithm searches for a path recursively using a child node
30   each time until reaching a predetermined depth, at which point it retraces it's steps and tries other child nodes. A best-first search chooses paths preferentially on the basis of an estimate of the shortest remaining distance to the goal.

The document discusses the various merits and disadvantages of search algorithms,
35   including the preferred search algorithm known as A*, which provides an optimal method (both in terms of processing requirements and quality of path) for many types of pathfinding

problems. The document proposes different techniques for overcoming some of the disadvantages in the A* algorithm. These include how to select an estimate of the remaining distance to the goal, and how the processing requirements can be reduced by splitting the virtual world into different regions (which can then be calculated separately) and using more
5 efficient data structures.

The article entitled *"Middleware Solutions for Artificial Intelligence in Computer Games"* (Chapter 3 – AI Techniques) by Jan-Harald Fredriksen of the Norwegian University of Science and Technology, located at http://www.idi.ntnu.no/grupper/su/fordypningsprosjekt-
10 2003/fordypning2003-Jan-Harald-Fredriksen.pdf is also concerned with pathfinding techniques in virtual worlds, and the use of the A* algorithm. This document considers the use of cost constraints for different terrains to be taken into account when calculating a path through the virtual world, and the use of hierarchical pathfinding (finding a broad overall route before computing the detailed portions) and portals (which split regions of the virtual world
15 into smaller parts) for improved computation performance.

The underlying structure of the virtual world on which the pathfinding in this document is based has two alternative versions. The first is referred to as the POV (points of visibility) approach, in which a plurality of nodes (called waypoints) are introduced into the virtual
20 world, and which the path is found on the basis of line-of-sight links between the nodes. The second approach is the use of a navigation mesh in which a plurality of convex polygons cover the surface of the world, and in this case the pathfinding algorithm considers each polygon to be a node and calculates a path through the virtual world accordingly.

25 In addition, there currently exist systems for automating the plotting of travel routes through parts of the real world. A good example of such a system is the "Route Planner" tool provided on the web-site of The AA organisation (see http://www.theAA.com). Such systems take into account certain preferences of the user (eg to avoid motorways, shortest distance, shortest time, etc). However the present inventors are not aware of any systems which
30 enable routes to take account of less utilitarian preferences of users in anything less than a very rudimentary manner. Also, such systems are necessarily restricted to calculating a route along predefined paths (i.e. where roads exist), and are therefore restricted in the flexibility of the final route that is generated.

35 **SUMMARY OF THE INVENTION**

4

According to a first aspect of the present invention, there is provided a method of determining a path along some of a plurality of points in a representation of a virtual world, the representation comprising a plurality of initially defined points, and obstructions through which the path cannot pass, the method comprising:

5   (a) defining a plurality of additional points by repeating the following step:

               defining a new point located on a line between two existing points, wherein the line does not pass through any of the obstructions; and

         (b) calculating the path based on any combination of new and/or initially defined points

10

This invention is applicable to any type of virtual world, which might include for example fantasy worlds or worlds based upon a real environment such as a historic building or design for a future building, or a house that is for sale, and in which the obstructions might comprise walls or other obstacles. The representation in the sense used above is likely to refer to

15   stored digital data that represents such a world.

The new point may be located on the mid point of the link. Step (a) may further comprise deleting the new point if it is less than a predefined distance from another of the points, and the predefined distance may vary in different regions of the virtual world. The method may

20   further comprise identifying a link as not being suitable for providing the location for a new point if said link intersects another one of said links which is shorter. A new point may be deleted if it does not have line of sight to each of a pair of points which do not have line of sight to each other and/or may be deleted if it does not form part of a path between two other nodes that is shorter than the shortest path which would exist between said two points

25   without said new point.

Advantageously, the arrangements of the invention employ a technique which allows a content provider to define some initial points for navigation in a virtual world, allowing the content provider to influence the routes taken and the items of interest that can be visited by

30   a user. The system then automatedly multiplies the number of points in a manner which will improve the user's experience along the subsequently generated path through the environment compared with only a limited set of originally defined navigation points.

In comparison with previously known systems in which a path through a virtual world is not

35   constrained to specific navigation points, the embodiments of the invention provide improved processing during the actual pathfinding procedure. This is because no collision detection

(ensuring the path avoids obstacles, etc) is required during pathfinding in the embodiments since all the new navigable nodes and links have been added on line of sight principles, and therefore bydefinition already avoid all the potential obstacles.

5   Additionally, these embodiments provide advantages over previous known systems such as the AA Routefinder which are necessarily limited to finding routes along specified predefined roads, etc. In contrast, in the embodiments, the user is able to navigate within all areas of the virtual world.

10

According to a second aspect of the present invention, there is provided a system for storing a representation of a virtual world, the representation comprising a plurality of initially defined points, and obstructions through which the path cannot pass, and in which two points are said to have line of sight if a straight line link between the two points does not pass through
15   any of the obstructions, the system being arranged to determine a path along some of a plurality of points in said representation by

(a) defining a plurality of additional points by repeating the following step:
defining a new point located on a line of sight link between two existing points

20   (b) calculating the path based on any combination of new and/or initially defined points

BRIEF DESCRIPTION OF THE DRAWINGS
25

For a better understanding of the present invention, specific embodiments will now be described, by way of example, with reference to the accompanying drawings, in which:
Figure 1 shows a schematic of a system according to an embodiment of the invention;
Figure 2 shows a diagrammatic representation of a virtual world environment;
30   Figures 3 to 12 show further diagrammatic representations of the virtual world environment of Figure 2; and
Figure 13 shows a flow chart of the method for processing points within a virtual world in an embodiment of the invention.

35   **DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION**

Figure 1 illustrates a system 1 for creating, and allowing a user to navigate within, a virtual world. The system comprises a server computer 2, accessible to a content provider (a human user who will input the specification for the virtual world) via terminal 3, and accessible to a user (desiring to view and navigate through the virtual world) via another

5   terminal 4. In this particular embodiment, access from both terminals to the server 2 is through a general network 5, such as for example the internet, although it is understood that the illustrated computer arrangement may be replaced by any suitable alternative which allows a content provider to design and implement a virtual world, which a user can subsequently access and interact with.

10

A schematic of a simple virtual world environment is shown in Figure 2. This represents a room 20, including walls 21, doors 22, and various obstacles 23, 24 and 25 within the room. When specifying the design of this virtual world 20, the content provider inputs data in the form of co-ordinates representing the shape of the environment into the server 2 via terminal

15   3. The data might include co-ordinates representing a collisionable mesh defining the obstacles 23, 24, and 25 through which a navigating user is not allowed to pass.

Feature Nodes

20   The content provider is able to define features of interest within the room that a user may wish to view. For example, if the virtual world represents a tour through a historic building, it may be desirable to insert features such as statues, furniture, paintings and information boards. These may be represented by objects within the room, for example statue object 26, painting 27, and information boards (obstacle 25).

25

In order that these features may be visited by a user, the content provider defines feature nodes 28A, 28B and 28C (shown in Figure 3) associated with the features, which the user can view or interact with. A feature node is therefore essentially a point in the virtual world, linked either to an object or to the collisionable mesh, and in the embodiment has the

30   following data set:

Node ID
Feature Group ID
Object ID of the linked mesh / object
XYZ of object pivot point data

| Sphere of influence settings |
| Display settings |

Node ID – an unique identifier for the node (e.g. text string "monet_painting_01" or numerical identifier "00563")

Feature Group ID – an identifier for the type of feature associated with the node, for
5   example all paintings may be in a first Feature Group identified as "paintings", whilst information boards might be a second Feature Group "information_points". The choice of groupings is highly dependent on the content provider and the type of features they wish to distinguish within the virtual world.

Object ID of linked mesh / object – this is the identifier for the object or collisionable mesh
10   associated with the feature node, and may be the same as the node ID (e.g. "monet_painting_01")

XYZ of Object Pivot Point Data – these are the co-ordinates giving the location of the feature node (and may be obtained by extraction from properties of the object / mesh with which the feature node is associated)

15   **Sphere of influence settings** – these define (for a 3D world) a three dimensional sphere (for example, illustrated as circle 31 in Figure 3) surrounding the feature node, and are used during the calculation of navigation paths through the virtual world

Display settings – these are some of the settings to be used for text / images associated with the feature

20

Navigational Nodes

The content provider also initially defines a plurality of navigational nodes (29A, 29B, 29C, etc) within the virtual world, as shown in Figure 3. Essentially, these provide a simple matrix
25   of navigational points which the automated path generation algorithm can use to plot a route through the environment. Associated with each navigational node is the following data set:

| Node ID |
| Node Level ID (optional) |
| XYZ position |
| LoS data |
| LoS constraint settings (optional) |
| FoV constraint settings (optional) |

Orientation constraint settings (optional)

Node ID – a unique identifier for the node

Node Level ID – this is an identifier which allows a level for the node to be defined, effectively grouping the nodes into different subsets, and used during the path finding

5   process to split the virtual world into different regions for processing efficiency. For example, two different rooms in the virtual world could be represented by different regions of nodes, with the doorway between them comprising a switch node. For processing efficiency, a path can initially be calculated based on only the nodes of one of the rooms (thus saving erroneously calculating at this stage paths which enter the second room, and only later

10   allowing the pathfinding calculation to extend to the second room via the switch node). Additionally it is possible that the different sub-sets could cover overlapping regions (for example a first level of nodes might provide coarse coverage of a large area while different levels might provide finer coverage of the same area or subsets of that area).

XYZ position– these are the x, y, z co-ordinates giving the location of the navigational node

15   **LoS data** – this data field will be empty initially, but is later used to contain a list of all nodes which the present node has Line of Sight (LoS) to, i.e. all the nodes that it is able to link to (see later for further details)

LoS constraint settings – the content producer may choose to manually define an angular LoS constraint (e.g. an angle range, in which the node is only allowed to search for nodes to

20   link to if they fall within that viewing angle, see later for more details)

FoV constraint settings – the content producer may choose to define angular constraints for the Field of View (FoV) which will constrain the range of images allowed to be displayed to a user at that node (e.g. for node 29F, the FoV may be constrained in the angular direction towards statue object 28A to ensure that the user sees the object)

25   **Orientation constraint settings** – for some navigational nodes, the content producer may choose to constrain the allowed orientation of the user as they approach the node, by defining orientation constraint settings this forces the calculated path to approach the node from a particular direction.

30   The content provider is thus able, through the use of navigational nodes, to define key positions within the virtual world from which the potential user is able to best experience the features of interest within the environment. The content provider can define the exact position, direction of approach (orientation settings), and viewing angle (FoV constraint settings) so as to increase the quality of the user's experience of the features of interest,

35   compared with other systems which are completely automated.

Processing

The subsequent processing is explained with reference to the flow diagram in Figure 13.

5   After node data has been input to the system (step 40), a node matrix is automatically
created for the virtual world (step 41). This identifies for each navigational node all the other
navigational nodes it is able to link to. Node links are calculated following basic rules of LoS
(line of sight) for two nodes. This takes into account the collisionable mesh and any other
objects deemed to crop the LoS of the nodes (i.e. no links are allowed to pass through

10   obstructions because a path between nodes cannot pass through these). Figure 4 illustrates
all possible LoS links for navigational nodes in the virtual world 20 (i.e. the lines linking nodes
29A, 29B, etc). For example, Node 29A has direct line of sight to both nodes 29B and 29F,
but not to 29C because this would pass through the obstacle 23. A node matrix for virtual
world 20 of Figure 4 is represented in tabular form below:

15

Node	Links to other nodes
29A	29B, 29F
29B	29A, 29C, 29D, 29E
29C	29B, 29D, 29F
29D	29B, 29C, 29E
29E	29B, 29D, 29F, 29G
29F	29A, 29C, 29E
29G	29E

At this stage each navigational node's data set is provided with LoS data on all its visible
partners, including their XYZ position. Any navigational node which does not have line of
sight to any other navigational nodes is deemed invalid and may be deleted from the system.

20

If the content provider had specified any LoS constraints for particular navigational nodes,
then this would crop the angle it is allowed to search for node links. This is controlled using
"LoS constraint settings" for the node, comprising degree values which slice the otherwise
allowed 360 degree LoS. In this case, the node matrix above would differ as some of the

25   links shown there would not be allowed. This functionality allows the content provider to
create channels of nodes that ignore other nodes around them, and to produce breaks in the
node matrix where nodes are not desired. This can improve the user experience because in

this manner the content provider can influence the routes taken by a user within the virtual world.

The next stages of processing will add in additional nodes on the basis of the LoS links
5   already defined. However, before that occurs, the system checks all the links (step 42) to decide which links will / will not be used as a base for the new nodes. In particular, the system looks for any crossed-links (such as the 29B-29E link which crosses the 29C-29D link), and marks the longest of the crossed links as not being available for use as a base for the new nodes. In this embodiment a data flag is associated with the link (in either the node
10   matrix or the individual nodes' LoS link data) to indicate that this link is "not to be subdivided". The affected link is shown as a dotted line in Figure 5.

The next stage of processing involves the automatic creation of a more dense matrix of nodes (step 43). This is achieved by subdividing the LoS links between the existing
15   navigational nodes so as to add new navigational nodes (also referred to as subdivided nodes) at each midpoint (step 44). Figure 6 shows these subdivided nodes (30A, 30B, 30C,... etc). As more and more new nodes are added during each cycle through step 43, the density of nodes increases, up to a maximum allowed node density. This density may be predefined in advance by the content provider, for example by specifying a minimum allowed
20   distance from a node to it's nearest neighbour. After each new node is added, the system checks whether the new node meets the density criteria (step 45). If it falls outside the allowed density criteria, the new node is deleted (step 46). Otherwise, the node is kept. Once every LoS link has been examined for subdivision (step 47), the system checks whether any new nodes were added during the last pass (step 48). If no new nodes were created during a
25   pass, the system terminates (step 49).

For each pass, after the new nodes have been added, the LoS data for every node (both the navigational nodes and each newly created subdivided node) is updated to take account of all the multiple new possible LoS links (step 50). Figure 7 illustrates the newly created dense
30   matrix with many more possible LoS links between the nodes.

The system now checks which of the newly added nodes are valid (step 51). A node is defined as valid if it links any two other nodes which do not otherwise have line of sight to each other. This is illustrated for node 30A in Figure 8. In this drawing, for clarity, the only
35   nodes and links illustrated are node 30A and the seven other nodes it links to. It can clearly be seen that node 30A acts as a bridge for many different pairs of nodes (i.e. it forms a

connection between two nodes which do not have LoS to each other), such as between nodes 29B and 30F. Therefore node 30A is valid, and in fact only requires that it is connected to one pair of nodes which do not have LoS, to be defined as a valid node. Any invalid nodes are deleted.

5

Figure 9 shows the same procedure for node 30B. Node 30B is defined as valid. However at this stage, any of 30Bs links which do not assist in bridging between two nodes are deleted. This is the case for the link between node 30B and 29C. Since node 29C already has line of sight links to all the nodes shown on Figure 9 (and reciprocally node 30B also has line of

10 sight to all the nodes linked to 29C), therefore link 30B-29C cannot help to bridge between any pair of nodes that do not have line of sight to each other, and the link 30B-29C is deleted.

Link 30B-30C would also appear from Figure 9 to be redundant, because node 30C already

15 has line of sight to all the nodes in Figure 9. However, link 30B-30C is actually valid because it bridges, for example, nodes 30B and 30A (which do not have line of sight to each other) which can be seen from Figure 7, and therefore link 30B-30C would not be deleted.

Validity checks are carried out in this manner for all new nodes, and two others (30E and

20 30H) are shown, in Figures 10 and 11 respectively.

After the validity checks have been performed on the new nodes, the system again checks for any crossed links (step 52). These are the links which are not to be used in any subsequent cycles of subdivision. The links are identified as before, by looking for links which

25 cross each other and flagging the longest ones as "not to be subdivided". This has been performed for all the links, and the resulting link matrix is illustrated in Figure 12 (where dotted lines illustrate those links which are not to be used in the future for subdivisions to add more nodes).

30 Thus one cycle of subdivision has been completed, and even after this, the number of nodes has more than doubled, and the connectivity between the nodes has greatly increased. For example node 29D previously only had links to 29B, 29C and 29E, but is now additionally linked to 30B, 30C, 30E and 30F.

35 Further cycles of subdivision (step 43) may be repeated, until the system has created a suitably dense matrix of nodes which map out the non-collisionable environment in the virtual

world. Advantageously, during the subsequent pathfinding calculation, there is no need to calculate collision data in real time because the matrix nodes are all constrained to navigable areas. In addition, with the higher density of nodes in the environment, the system can generate far more efficient paths than with the originally defined node matrix, and which

5   enhance the user experience by giving a much smoother path through the virtual world which typically follows the type of behaviour a user is used to when navigating a real world environment. Once the path has been calculated a spline is generated to further interpolate and hence to smooth the journey still further. In the present embodiment, once the spline has been generated along a particular path, a default speed along the path is set. In the

10   present embodiment this is set to vary between a fairly fast speed in areas of little interest and to a relatively lower speed around features of interest. Ideally this is done by smoothly varying the speed to have minimum values at the feature nodes of most interest and to have maximum speeds in between these points.

15   Although in the embodiment above, the node density criteria is specified in terms of the minimum allowed distance between a new node and an already existing node, this could be replaced by any other suitable mechanism for specifying and testing the node density. Another possible method is to specify in advance a maximum allowed number of cycles of subdivision before the algorithm must terminate.

20
Although In the embodiment above, the check for whether a new node is valid involves checking whether the node acts as a bridge between two other nodes, an alternative/additional criteria which may be used to define whether each newly added node is valid is to check whether the new node assists in providing a shorter path between any pair

25   of nodes.

Pathfinding

As mentioned earlier, the A* algorithm is an optimal pathfinding algorithm both in terms of the

30   processing time required, and the quality of the path found. Start and destination nodes are defined in advance, and the algorithm explores multiple different paths to find the best route from the start to the destination. Consider the case of a user who wishes to tour through virtual world 20, entering at door 22, and who has indicated via a suitable interface that he desires to approach information board 25.

35

The start node is defined as node 29A, and the destination node as 29G. The A* algorithm operates by maintaining two lists of nodes: a list of nodes that have already been explored (the Closed list) and a list of nodes linked to the ones that have been explored but have yet to be explored themselves (the Open list). The algorithm takes a node from the Open list,

5   and if it is the destination node then the algorithm terminates. If not, all the nodes linked to the chosen node are added to the Open list. The A* algorithm is known as a heuristic method because it uses estimates to guide the search for nodes. In particular, in order to decide which node should be selected next from the Open list, all the nodes have an evaluation function **f = cost + heuristic**, and the node which is selected as the next current node from

10 the Open list is the one with the lowest value for **f**.

The **cost** is a measure of the quality of the path so far from the start node to the current node (i.e. for a simple A* algorithm this could be the sum of the lengths of all the links in the path between the start node and the node). The **heuristic** function is an estimate of the cost to

15 get from the current node to the destination node. The results of the A* algorithm are very much dependent on the choice of cost and heuristic functions, and it is the specific weighting and cost values used in the evaluation function for the embodiment of the invention that allow the optimum path to be generated.

20 The **cost** function of the embodiment takes into account various factors of the route, together with any user specific preferences. The cost function is a weighted sum of the following form:

cost = w1 * distance + interest_component + w3 * angle

25

where **distance** is length of the links so far, and **w1** is the weighting ascribed to the current user indicating how important it is to them that the path is as short as possible. **Interest_component** gives a measure of how interesting the node (s) so far have been (but since we are looking to minimise the cost function, the lower the interest_component value

30 the more interesting the nodes so far have been). Thus, in the present embodiment a summation over the node or nodes included in the partial path built thus far of the form:

$$\sum_{PARTIAL\_PATH} \left(1 - w2 * Interest\right)$$

is used, in which w2 is the percentage weighting interest given to the feature group

35 associated with the node in question, and Interest is a measure of how strongly a node is

associated with the corresponding feature associated with the node. Thus, in the present embodiment, feature nodes (typically of great interest to a user) may be given an Interest value 1, whilst navigational nodes have interest value of 0 and therefore when summed as above give a higher cost. As mentioned, weighting **w2** gives a measure of how important it is to the user that they visit features of interest of each respective feature group.

Finally, **angle** gives a measure of how much the path so far wiggles, since this represents a sum (or average) of how much the angle between the entry and exit links for each node deviates from a straight line. Weighting **w3** gives a measure of how important it is for the user that the path wiggles as little as possible (i.e. a zero value of w3 would suggest that the user did not mind how wiggledy the path was whilst a high value (eg 100%) would indicate that the user would prefer to avoid wiggledy paths).

The **heuristic** is an estimate of the cost to get from the current node to the destination node, and one possible choice for this is:

$$\text{heuristic} = \text{euclidian\_distance\_to\_destination}$$

The heuristic in the embodiment is therefore based on the Euclidian distance to destination (i.e. a straight line from the current node to the destination node), although it could alternatively also include some measure of the distance, interest and angle weightings **w1**, **w2** and **w3** if desired.

The system therefore runs through the A* algorithm for the nodes using the evaluation function, until the final destination node is reached. Importantly, the weightings **w1**, **w2**, and **w3** can be varied depending on the user profile.

Defining a user profile

A suitable interface (not shown) allows the user to define their profile (e.g. what they are particularly interested in, and how they like to navigate through the virtual world). The user profile settings are then incorporated into the weightings above used for the A* algorithm, which consequently will result in different paths being generated for different users. The translation of the user interests into weightings allow different routes to be compared against each other such that the optimum path for that user can be generated. It can be seen that for

a user who is particularly interested in visiting features of interest (paintings, landmarks, etc) it is likely to be the case that the system will not calculate the quickest or shortest route between two points. Rather, the lowest cost path (i.e. the best route for that user) will be one which visits as many features of interest as possible.

5

The user may input their profile upon first entering the virtual world. Some options include:

Always select shortest path	on / off
Always select quickest path	on / off
Ignore all features	on / off
Gaze orientation	on / off
Velocity control	on / off
Importance of path being short (w1)	weighting
Importance of path not being wiggly (w3)	weighting
Importance of visiting Feature Group 1 [e.g. paintings] (w2)	weighting
Importance of visiting Feature Group 2 [e.g. information points] (w2)	weighting
Importance of visiting Feature Group 3 [e.g. landmarks] (w2)	weighting

The first five options are overrides, and are thus non-weighted. For example, if the user
10 selected "Always select shortest path" as ON, then the weighting **w2** for the features of interest become irrelevant and will be set to zero.

The final five options are all weighted, and however the weightings are distributed they should all be normalised (e.g. add up to 100% in total). Any option which is not important to
15 the user and not given a weighting is deemed irrelevant and not taken into account in the pathfinding process.

Dynamic Navigational Nodes

20 In addition to the fixed navigational nodes (29A, 29B, 29C, etc), defined initially by the content provider, the system may also include dynamic navigational nodes (not shown). These nodes operate like navigational nodes in the sense that they can form part of the path generated for the user, but provide enhanced functionality because the user is able to move them. A dynamic navigational node is associated with an interactive mesh object in the
25 virtual world environment that can be moved around by the user. The user is able to specify

that they want the dynamic navigational node to be a destination point, and the path will be calculated to it's location.

If a dynamic navigational node is placed a large distance from other navigational nodes, then
5   the subdivision process described earlier may be used to fill in new nodes until a suitable node density is achieved in that area. This then allows a smooth path to be calculated to the node.

If a dynamic navigational node is placed without a line of sight to any other navigational
10   nodes (i.e. in a blind spot) then the node is defined as invalid – lost from the Node Matrix (although still present in the environment) and the user is prevented from selecting it as a destination. Alternatively, the system may prevent the user from placing the node in a blind spot, and only allows it to be moved into valid positions.

15   **Switch Navigational Nodes**

In addition to the fixed navigational nodes and the dynamic navigational nodes, the system may also include switch navigational nodes (not shown), defined by the content producer. As mentioned earlier, navigational nodes have a Level ID, specifying which level the node
20   belongs to, and effectively grouping the nodes into different subsets for more efficient path calculation. For example, if the virtual world consists of two different rooms, then the navigational nodes in one room may be defined as belonging to one level, and the navigational nodes in the second room to a different level. During the pathfinding process in one room, only the nodes of that level are evaluated, saving takes into account nodes on a
25   different level. The switch nodes then provide the functionality to swap from one level to the second level (e.g. the switch navigational node might be positioned in a doorway between the two rooms).

Associated with each switch navigational node is the following data set:
30

Node ID
Node Level ID #1
Node Level ID #2
XYZ position
LoS data

Switching data

Node ID – an identifier for the node

Node Level ID #1 – the identifier for the first level

Node Level ID #2 – the identifier for the second level

5   **XYZ position**– these are the x, y, z co-ordinates giving the location of the switch navigational node

LoS data – this gives the data for all nodes (for both levels) which the present node has Line of Sight (LoS) to

Switching data – gives the criteria of when the generated path can / cannot switch from

10   navigational nodes on one level to nodes on the other level

Event Feature Nodes

As described earlier, the content provider is able to define feature nodes (28A, 28B, 28C,

15   etc) associated with the features of interest in a virtual world. The content producer is also able to define event feature nodes (not shown) which have the extra functionality that they react to users within the virtual world, or are only active at certain predefined times.

These event feature nodes expand the functionality of the system into the gaming and

20   interactive online environment platforms, where multiple users can interact with the environment and with each other (e.g. the Event Feature may simulate an explosion, or the opening and closing of an interactive chat-room / race event. The content provider must specify when the event feature will be active / inactive, or what triggers it will respond to. An example data set is as follows:

25

Node ID
Feature Group ID
Trigger data
XYZ of pivot point data
Sphere of influence settings
Time settings
Display settings

Node ID – an identifier for the node (e.g. text string "explosion")

Feature Group ID – an identifier for the type of feature associated with the node

18

Trigger data – the conditions which will cause the event feature node to become active

XYZ of Pivot Point Data – these are the co-ordinates giving the location of the event feature node

Sphere of influence settings – these define (for a 3D world) a three dimensional sphere

5   surrounding the feature node. However, in addition to the settings used for the standard feature nodes, these additionally include a time dependant feature which may allow the sphere of influence to degrade over time.

Time settings – these define when the event feature node will be active / inactive

Display settings – these are the settings to be used for text / images associated with the

10   feature when constrained

Further functionality

15   As explained above, the processing and pathfinding stages result in a spline path which will be used to take the user through the environment. The spline path therefore comprises a plurality of nodes each joined by links to the next node. The system applies smoothing to this path by interpolating the spline so as to improve further the user experience.

20   During real time display, when the user is guided through the environment they have the further option to control their field of view. As discussed earlier, the content provider can specify particular angles of viewing (FoV constraints) which are to be applied for specific feature nodes so as to ensure the user does not miss features of interest. The user can choose to toggle control to either operate this automatic field of view presentation or

25   alternatively to control manually their own field of view.

In the embodiments described above, it is explained that the processing stage which creates the more dense matrix of nodes occurs after the content provider has defined the contents within a virtual world, but before the system is used by a user desiring to navigate within that

30   world. It is of course alternatively possible for either part or all of the processing stage which adds these subdivided nodes to be carried out in real time whilst the system is in use by a user. Indeed, an alternative arrangement is particularly desirable when the virtual world contains a number of dynamically moving objects (for example, a virtual environment of a city might include moving cars, buses, etc). In this dynamic case, only some subdivision (for the

35   regions of the world containing static objects such as buildings, etc) could be carried out in advance before a user accesses the system, with the remaining processing to add the extra

nodes occurring each time a user specified that they desire a path to be created. The later processing stage thus takes into account the positions of these dynamic objects at the instant the user desires the path to be created. This functionality could be supplemented by defining some regions in the world in which this later (dynamic) subdivision will occur (for

5   example, streets within a virtual city), and other regions in the world when the dynamic subdivision will not occur because they typically contain only stationary objects (for example, the insides of some buildings). This is a typical example when switch nodes (mentioned above) might be used to delimit the node links between different regions.

10   Furthermore, it may be necessary in real time to create new nodes corresponding to the start and destination points selected by a user for a path generation procedure. If nodes do not already exist at these locations then the system may be used to automatedly create such nodes, and additionally use processing to add new subdivided nodes in only the regions close to the start and destination nodes so as to link them to the already existing node

15   matrix.

CLAIMS

5   1.   A method of determining a path along some of a plurality of points in a representation of a virtual world, the representation comprising a plurality of initially defined points, and obstructions through which the path cannot pass, and in which two points are said to have line of sight if a straight line link between the two points does not pass through any of the obstructions, the method comprising:

10   (a) defining a plurality of additional points by repeating the following step:

defining a new point located on a line of sight link between two existing points

          (b) calculating the path based on any combination of new and/or initially defined points

15

   2.   A method according to claim 1, wherein the new point is located on the mid point of the link.

   3.   A method according to claim 1 or 2, further comprising in (a) the step of:

20   deleting the new point if it is less than a predefined distance from another of the points.

   4.   A method according to claim 3, wherein the predefined distance varies in different regions of the virtual world.

25

   5.   A method according to any preceding claim, further comprising identifying a link as not being suitable for providing the location for a new point if said link intersects another one of said links which is shorter.

30   6.   A method according to any preceding claim, further comprising deleting a new point if it does not have line of sight to each of a pair of points which do not have line of sight to each other.

   7.   A method according to any of claims 1 to 5, further comprising deleting a new point

35   if it does not form part of a path between two other nodes that is shorter than the shortest path which would exist between said two points without said new point.

8.   A method according to any preceding claim, further comprising receiving an indication from a user that they desire a path to be calculated, and wherein said step (a) occurs after said indication has been received.

9.   A method according to claim 8, in which the virtual world comprises a plurality of regions each containing one or more points, and in which the step (a) is performed in relation to one or more of the regions prior to receiving said indication, and step (a) is performed in relation to other one or more of the regions after said indication has been received.

10.   A method according to claim 9 in which said other one or more regions contain obstacles which move within the virtual world.

11.   A method according to claim 9 in which said other one or more regions contain a point which has been created by the user as representing a start or destination point for said path.

12.   A system for storing a representation of a virtual world, the representation comprising a plurality of initially defined points, and obstructions through which the path cannot pass, and in which two points are said to have line of sight if a straight line link between the two points does not pass through any of the obstructions, the system being arranged to determine a path along some of a plurality of points in said representation by

        (a) defining a plurality of additional points by repeating the following step:
                defining a new point located on a line of sight link between two existing points
        (b) calculating the path based on any combination of new and/or initially defined points

13.   A system according to claim 12, wherein the new point is located on the mid point of the link.

14.   A system according to claim 12 or 13, further arranged in (a) to:
                delete the new point if it is less than a predefined distance from another of the points.

15. A system according to claim 14, wherein the predefined distance varies in different regions of the virtual world.

5   16. A system according to any of claims 12 to 15, further arranged to identify a link as not being suitable for providing the location for a new point if said link intersects another one of said links which is shorter.

17. A system according to any of claims 12 to 16, further arranged to delete a new point
10   if it does not have line of sight to each of a pair of points which do not have line of sight to each other.

18. A system according to any of claims 12 to 16, further arranged to delete a new point if it does not form part of a path between two other nodes that is shorter than the
15   shortest path which would exist between said two points without said new point.

19. A system according to any of claims 12 to 18, further arranged to receive an indication from a user that they desire a path to be calculated, and wherein said step (a) is performed after said indication has been received.
20

20. A system according to claim 19, in which the virtual world comprises a plurality of regions each containing one or more points, and in the system is arranged to perform step (a) in relation to one or more of the regions prior to receiving said indication, and to perform step (a) in relation to other one or more of the regions
25   after receiving said indication.

21. A system according to claim 20 in which said other one or more regions contain obstacles which move within the virtual world.

30   22. A system according to claim 20 in which said other one or more regions contain a point which has been created by the user as representing a start or destination point for said path.

23. A storage medium carrying computer readable code representing instructions for
35   causing one or more processors to perform the method according to any of claims 1 to 11 when the instructions are executed by the processor or processors.

24.   A computer program comprising instructions for causing one or more processors to perform the method according to any of claims 1 to 11 when the instructions are executed by the processor or processors.

5

25.   A computer data signal embodied in a carrier wave and representing instructions for causing one or more processors to perform the method according to any of claims 1 to 11 when the instructions are executed by the processor or processors.

10   26.   A storage medium carrying computer readable code representing instructions for causing one or more processors to operate as the system according to any of claims 12 to 22 when the instructions are executed by the processor or processors.

27.   A computer program comprising instructions for causing one or more processors to
15   operate as the system according to any of claims 12 to 22 when the instructions are executed by the processor or processors.

28.   A computer data signal embodied in a carrier wave and representing instructions for causing one or more processors to operate as the system according to any of
20   claims 12 to 22 when the instructions are executed by the processor or processors.

24

ABSTRACT

5

PATHFINDING SYSTEM

A computer system is arranged to automatically calculate a path along nodes in a virtual
world. After the co-ordinates for the virtual world environment have been initially defined (40),

10   including nodes along which a path may travel, the system automatically increases the
density of nodes in the environment up to a desired density. New nodes are added between
each pair of nodes which have line of sight to each other (44 to 47), dramatically increasing
the number of available links and nodes. This is repeated until a sufficient density of nodes is
reached and no more nodes are being added (48 and 49).

15

[Figure 13]

20

Figure 1
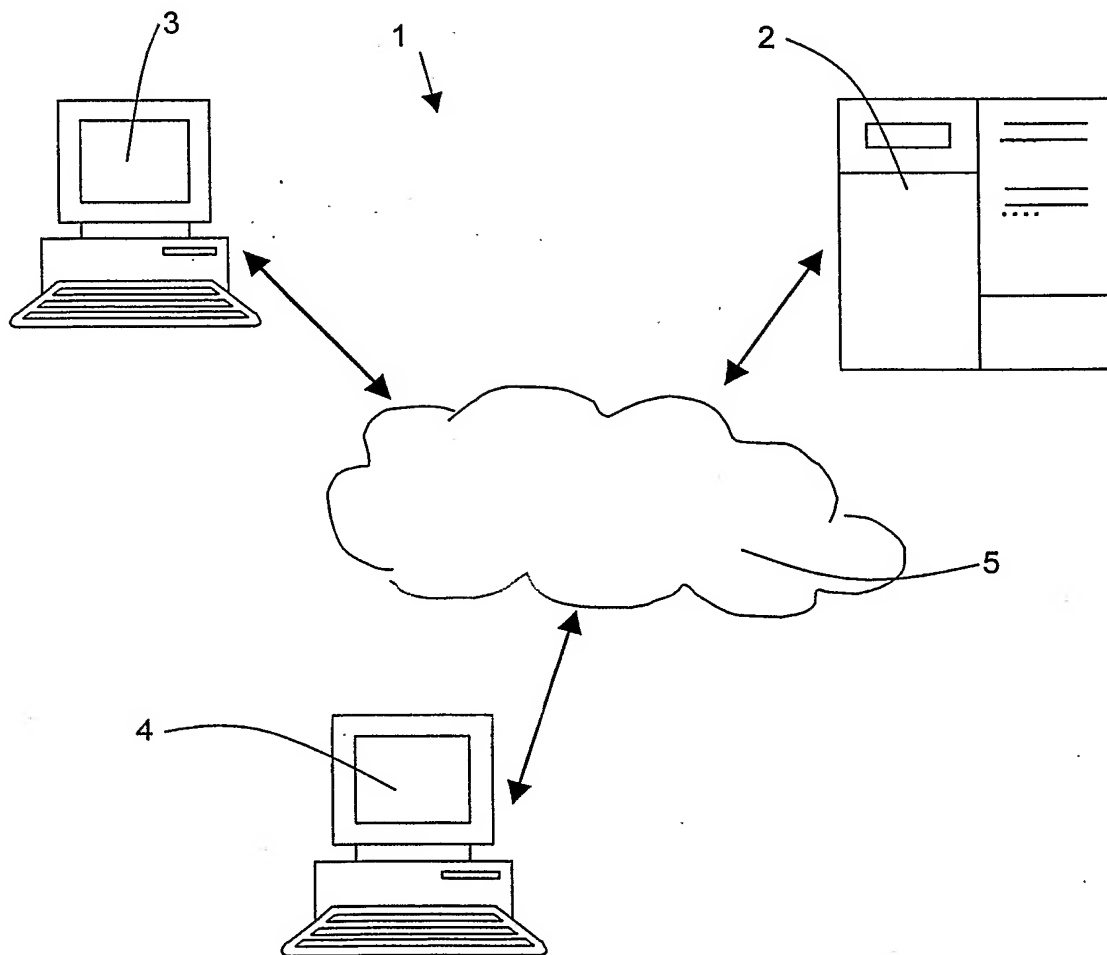
Figure 2
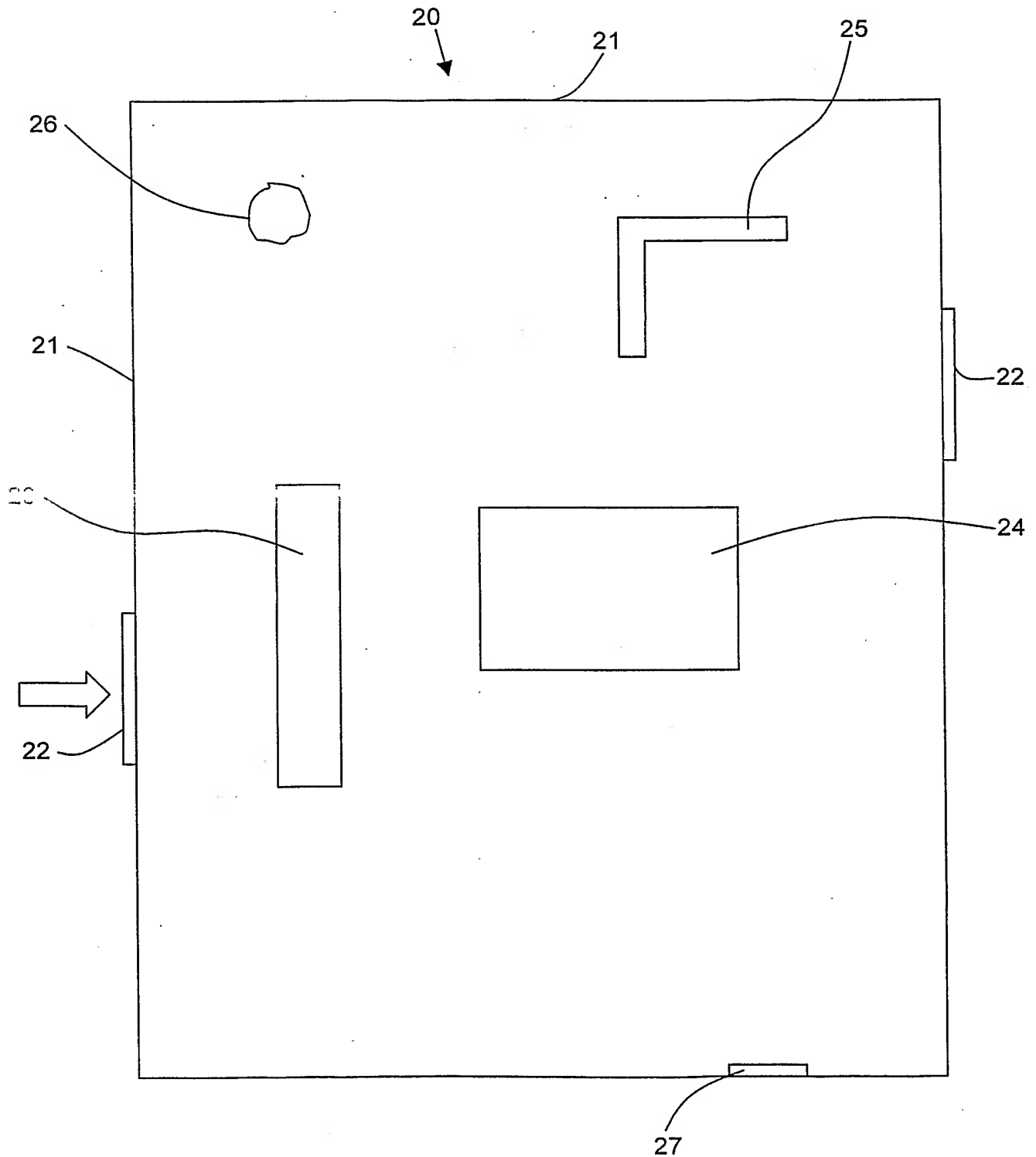
Figure 3
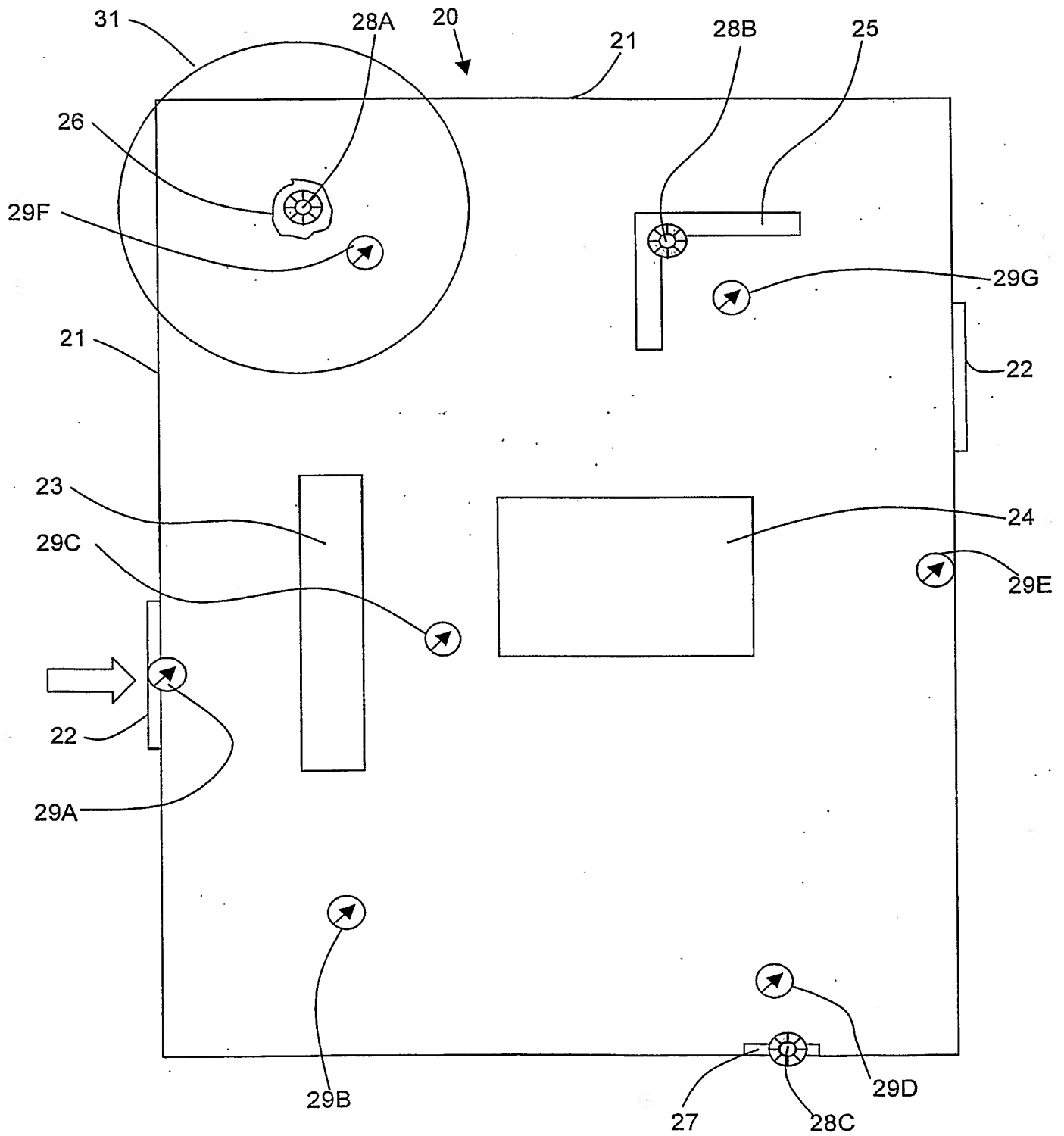
Figure 4
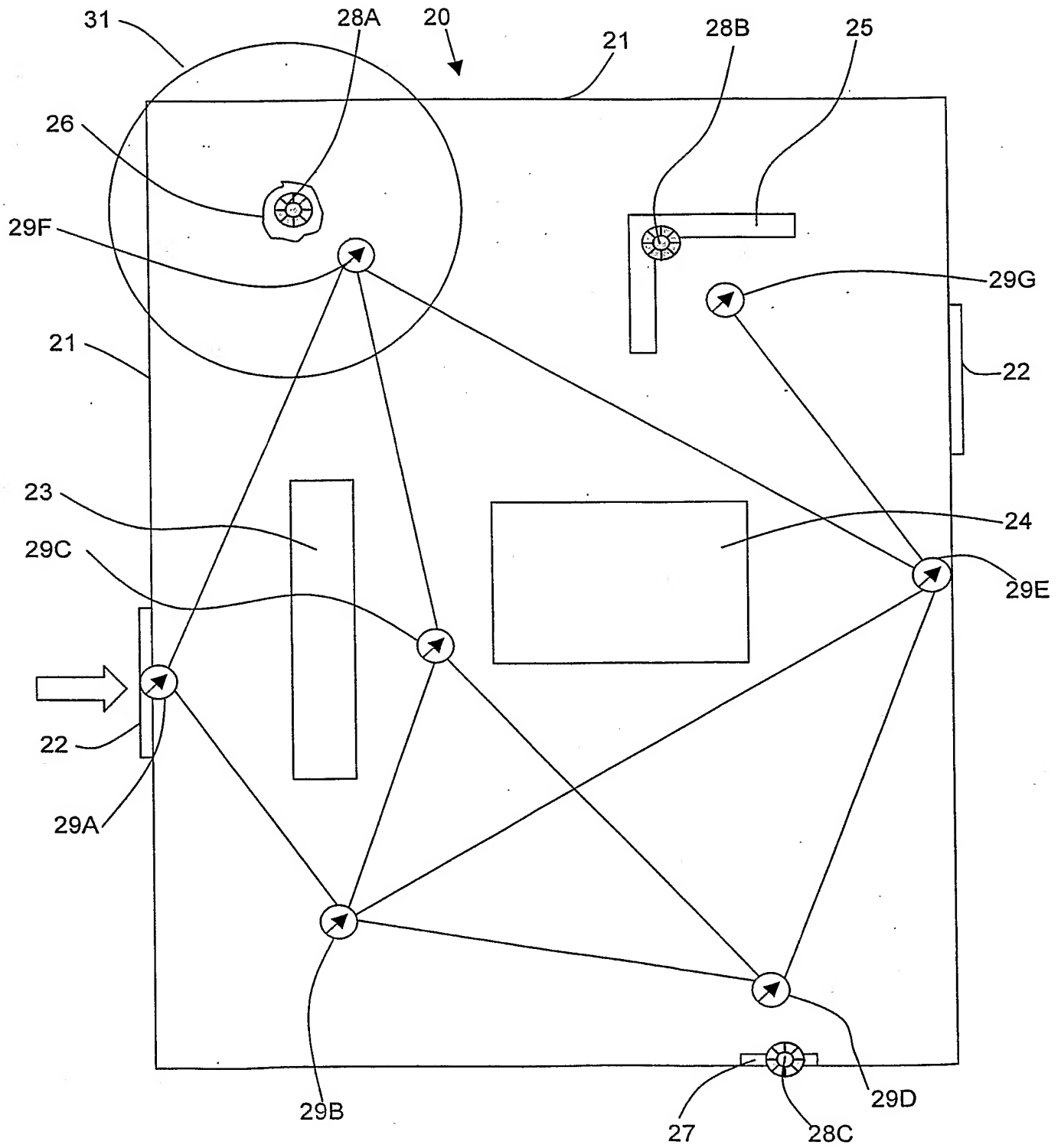
Figure 5

Figure 6

Figure 7

31   28A   20   30H   21   28B   25

26

29F

30I

21

30F

23

29C

22

29A

30A

30B

29G

22

30G

24

29E

30E

30D

29B   30C   27   28C   29D
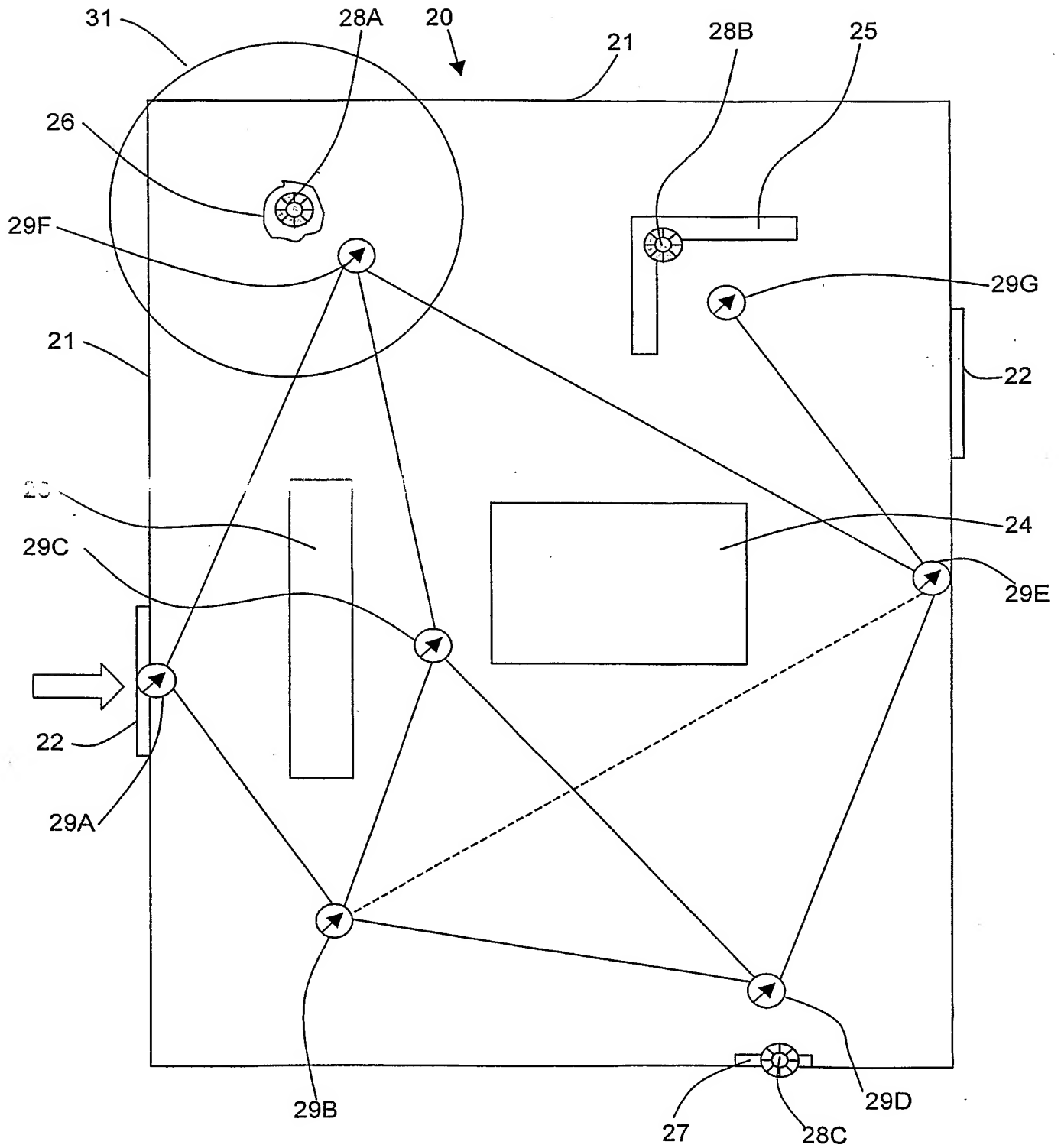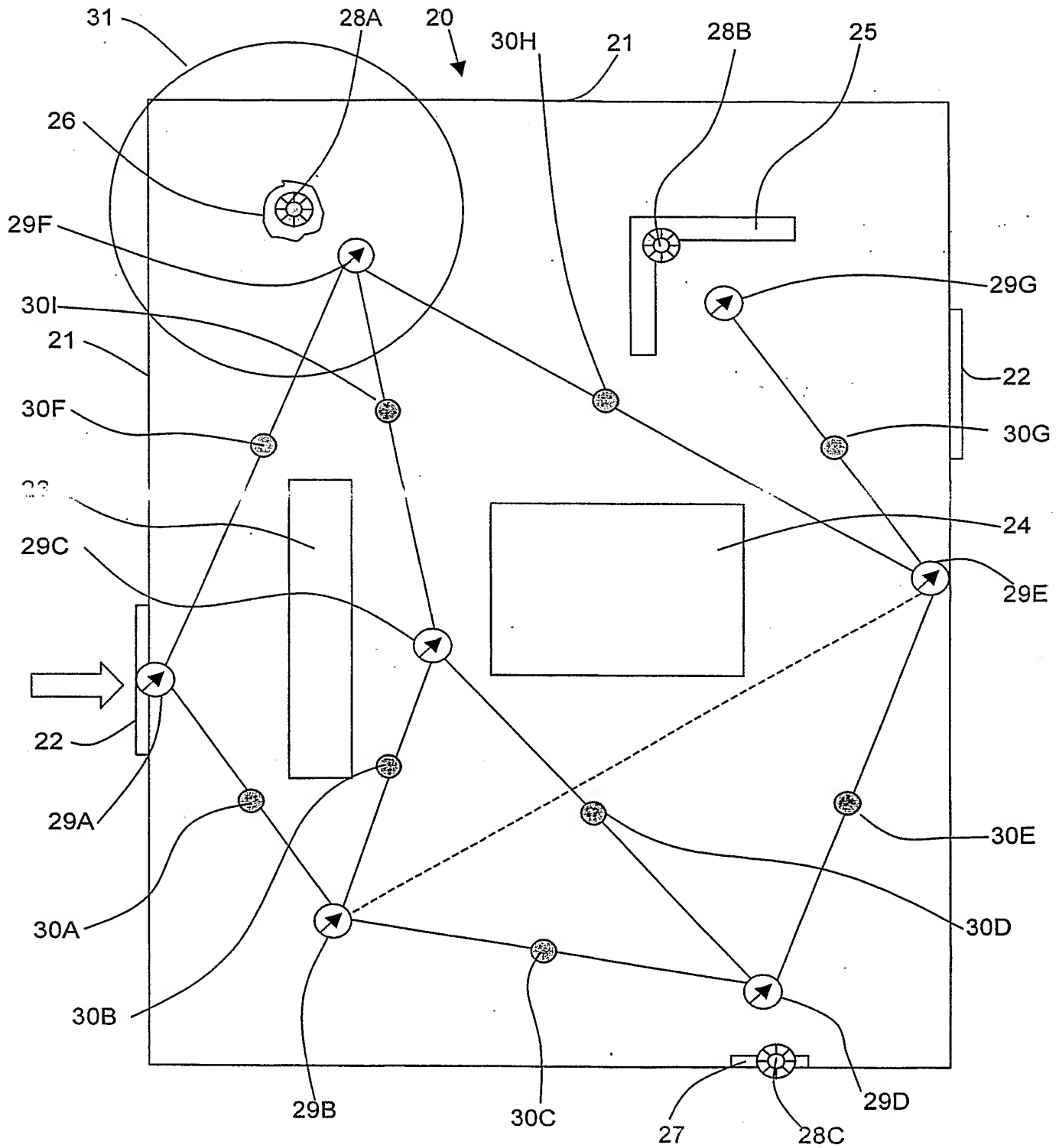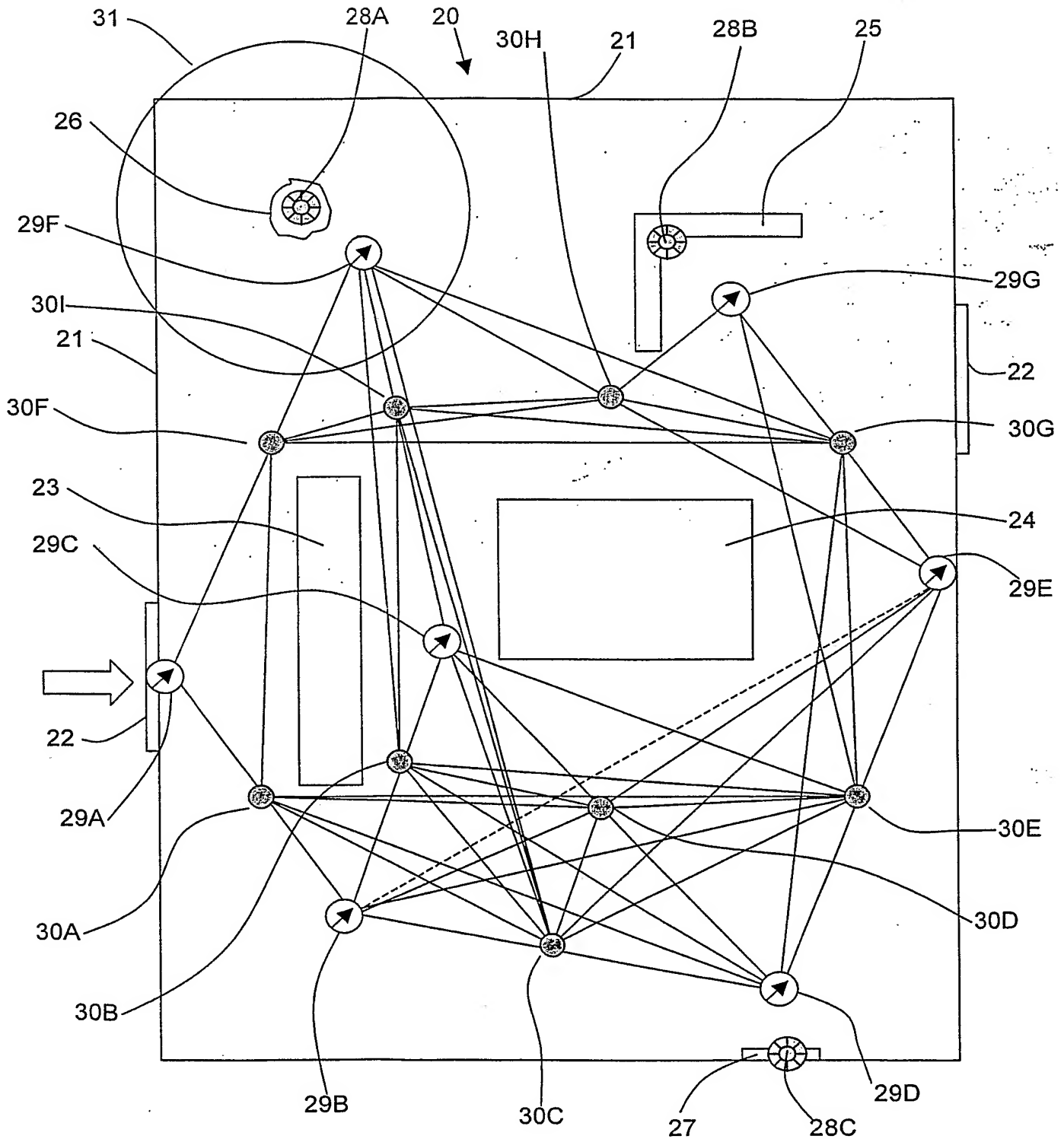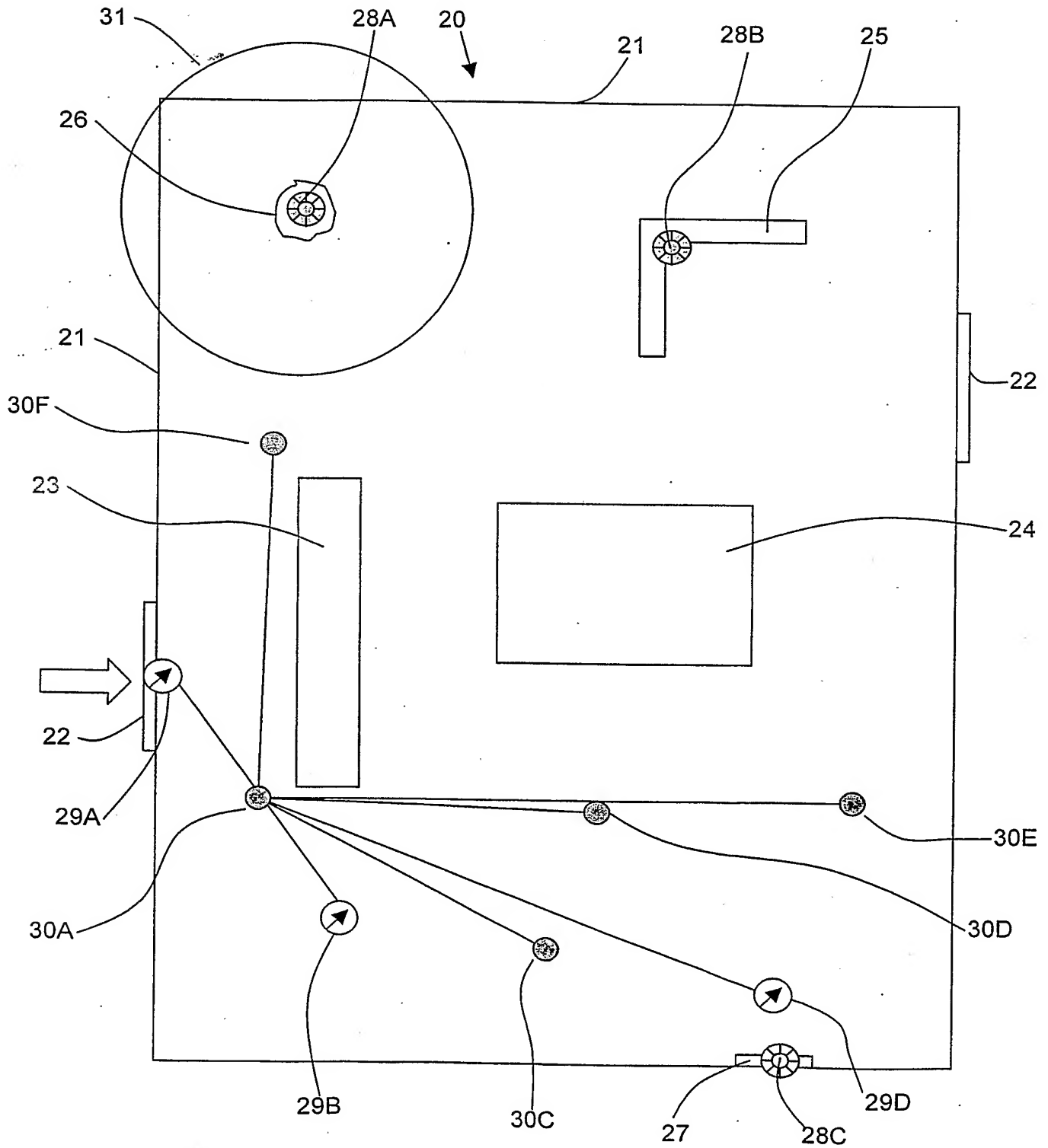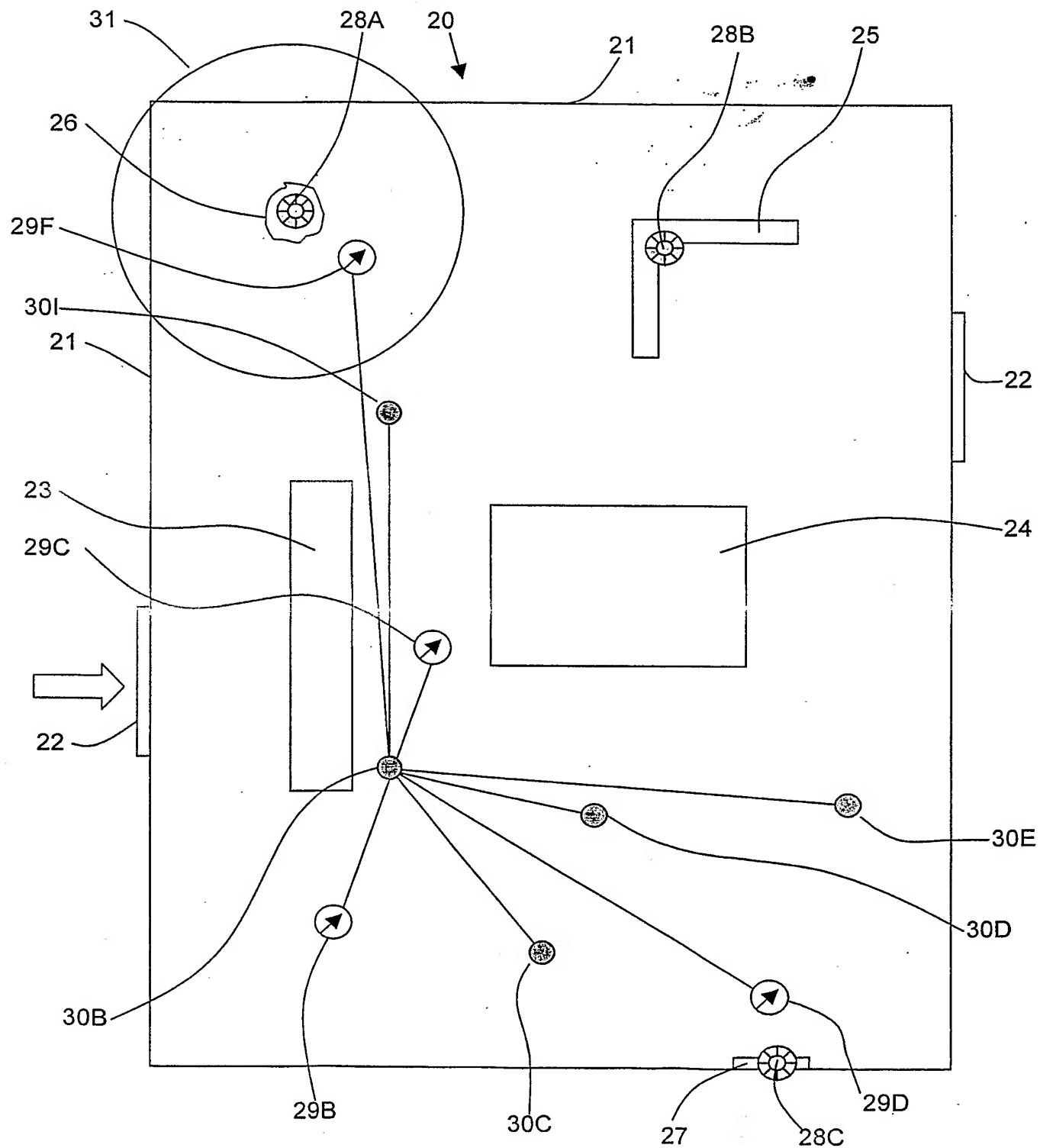
Figure 8

Figure 9

Figure 10
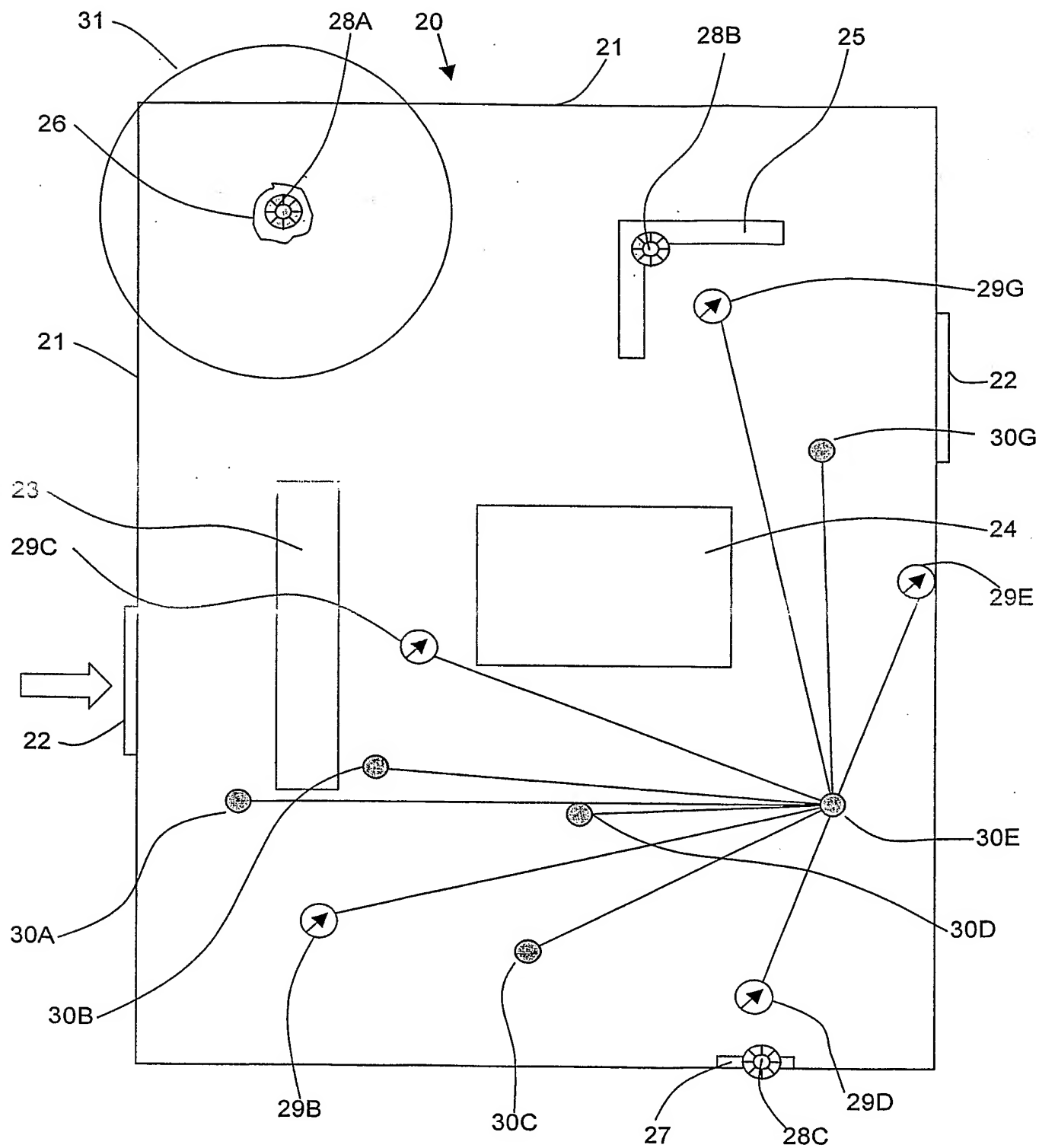
Figure 11

Figure 12

Figure 13

40 — content provider inputs virtual world, including navigational nodes

41 — node matrix created - defines links between each pair of navigational nodes in line of sight

42 — identify crossed links "not to be subdivided"

43

44 — examine next link, insert new node at midpoint of link

45 — does new node meet density criteria?

YES   NO

46 — new node is deleted

47 — are there still links unexamined on this pass?   YES

NO

48 — were any new nodes added during last pass?   NO

YES

49

terminate subdivision

50 — update line of sight links

51 — check new nodes are valid

52 — identify crossed links "not to be subdivided"